Claim Amendments

1. (Original): A method for keeping two independent busses coherent comprising:

writing data from an Input/Output (I/O) controller to a memory, the I/O controller sending the data to the memory via a first bus connected to a first port of a memory controller and the I/O controller;

sending a tag, from the I/O controller, after the data via the first bus through the first port, the tag being received by the memory controller;

requesting completion status of the data write from the I/O controller by a processing unit, the request being sent to the I/O controller via a second bus connected to a second port of the memory controller and the I/O controller; and

waiting for a tag acknowledgment, by the I/O controller, from the memory controller before providing notification to the processing unit that the data write has completed,

wherein the first bus and the second bus are coherent.

2. (Original) The method according to claim 1, comprising writing the data to the memory from one of at least one Direct Memory Access (DMA) controller at the I/O controller.

3-4. (Canceled).

5. (Original) The method according to claim 2, further comprising sending the data from the one of at least one DMA controller to a second memory at the I/O controller and then the writing the data to memory.

6. (Original) The method according to claim 5, further comprising sending the data, by the I/O controller, from the second memory via the first bus to a third memory at the memory controller and then the writing the data to memory.

7-9. (Canceled).

10. (Original) A system for keeping two independent busses coherent comprising:

at least one memory device;

a memory controller operably connected to the at least one memory device;

a processing unit operably connected to the memory controller; and

an Input/Output (I/O) controller operably connected to the memory controller by a first bus and a second bus, the I/O controller writing data to the at least one memory device via the first bus and the memory controller, the I/O controller sending a tag after the memory write to the memory controller via the first bus, the processing unit requesting status from the I/O controller via the memory controller and the second bus,

wherein the I/O controller waits for a tag acknowledgment from the memory controller before providing notification to the processing unit via the second bus that the data write has completed ensuring that the first bus and the second bus are coherent.

11. (Original) The system according to claim 10, further comprising a second memory at the memory controller and a third memory at the I/O controller, the data

write sent from the I/O controller to the memory via the third memory, first bus, and second memory.

12-23. (Canceled).

24. (Original) A system for keeping two independent busses coherent comprising:

at least one memory device;

a memory controller operably connected to the at least one memory device;

at least one processing unit operably connected to the memory controller;

at least one Input/Output (I/O) controller;

at least one first bus, one associated at least one first bus operably connected between one at least one I/O controller and the memory controller;

a second bus operably connected between the memory controller and each at least one I/O controller, each at least one I/O controller writing data to the at least one memory device via the associated at least one first bus and the memory controller, each at least one I/O controller sending a tag after the memory write to the memory controller via the associated first bus, the processing unit requesting status from each at least one I/O controller that initiates the write via the memory controller and the second bus,

wherein each at least one I/O controller waits for a tag acknowledgment from the memory controller before providing notification to the processing unit via the second bus that the data write has completed ensuring that each at least one first bus and the second bus are coherent.

25. (Original) The system according to claim 24, further comprising a second memory at the memory controller and a third memory at the I/O controller, the data write sent from the I/O controller to the memory via the third memory, first bus, and second memory.

26-30. (Canceled)

31. (Previously Presented) A method of an I/O controller comprising

transferring data to a memory controller via a first bus;

receiving a notification from the memory controller via the first bus after transferring the data to the memory controller; and

providing a processor with a completion status via a second bus that is different than the first bus after receiving the notification from the memory controller.

32. (Previously Presented) The method of claim 31 further comprising transferring a fence to the memory controller via the first bus that requests the memory controller to send the notification.

33. (Previously Presented) The method of claim 31 further comprising receiving from the processor via the second bus a status request for the data before providing the processor with the completion status.

34. (Previously Presented) The method of claim 31 further comprising receiving configuration information from the processor via the second bus that configures the I/O controller to write the data to the memory via the first bus and the memory controller.

35. (Previously Presented) The method of claim 31 further comprising

receiving configuration information from the processor via the second bus that configures the I/O controller to write data to the memory via the first bus and the memory controller,

transferring a fence to the memory controller via the first bus that requests the memory controller to send the notification, and

receiving from the processor via the second bus a status request for the data before providing the processor with the completion status.

36. (Previously Presented) The method of claim 31 wherein transferring the data further comprises transferring a tag that requests the memory controller to send the notification.

37. (Previously Presented) An I/O controller comprising

a first bus interface for a first bus,

a second bus interface for a second bus separate from the first bus, and

a direct memory access (DMA) controller

to receive via the first bus interface configuration information for a data write,

to write data to a memory via the second bus interface based upon the configuration information for the data write,

to send, after the data via the second bus interface, a fence that requests a notification of receipt of the fence, and

to generate a completion status for the data write based upon the notification.

38. (Previously Presented) The I/O controller of claim 37 wherein the DMA controller further transfers the completion status via the first bus interface in response to receiving a status request for the data write via the first bus interface.

39. (Previously Presented) The I/O controller of claim 38 wherein the DMA controller receives the status request prior to the notification and transfers the completion status after receiving the notification.

40. (Previously Presented) An apparatus comprising

a processor bus interface to receive configuration information for a data write from a processor bus,

a first I/O controller interface to transfer the configuration information for the data write to a first I/O controller bus,

a second I/O controller interface to receive data and a fence for the data write from a second I/O controller bus, and

a controller to write the data to a memory and to send a notification via the second I/O controller interface in response to the fence.

41. (Previously Presented) The apparatus of claim 40 wherein the controller sends the notification after writing the data to the memory.

42. (Previously Presented) The apparatus of claim 41 wherein

the processor bus further receives a status request for the data write from the processor bus,

the first I/O controller interface further transfers the status request and receives a completion status via the first I/O controller bus, and

the processor bus further transfers the completion status to the processor bus.

43. (Previously Presented) A system comprising

a processor

a memory controller coupled to the processor via a processor bus and adapted to write data to a memory, and

an I/O controller coupled to the memory controller via a first I/O controller bus and a second I/O controller bus, wherein the I/O controller

receives configuration information for a data write from the processor via the processor bus and the first I/O controller bus,

transfers, based upon the configuration information, data and a fence to the memory controller via the second I/O controller bus, and

generates a completion status for the based upon a notification received from the memory controller via the second I/O controller bus.

44. (Previously Presented) The system of claim 43 wherein the memory controller provides the I/O controller with the notification via the second I/O controller bus in response to receiving the fence via the second I/O controller bus.

45. (Previously Presented) The system of claim 44 wherein the memory controller provides the I/O controller with the notification after writing the data of the data write to the memory.

46. (Previously Presented) The system of claim 43 wherein the I/O controller receives a status request for the data write via the first I/O controller bus, and

waits for the notification from the memory controller before transferring, in response to the status request, the completion status on the first I/O controller bus.